|  |  |  |
| --- | --- | --- |
| RAZDA Co. | | |
| **Filename: [site\_status.py]** | | |
| **Summary***:*  The site\_status.py file is dedicated to monitoring and reporting on the health and performance of the Razda Market platform. It oversees critical components like the database, essential APIs, and server resources, providing real-time feedback and alerting admins of potential issues. Through periodic health checks and resource usage monitoring, this file ensures that the system runs optimally and that administrators are promptly informed of any problems. Integration with alert\_manager enhances its responsiveness by allowing alerts to be triggered automatically under critical conditions. | | |
| ***Processes*** | | |
| * **Database Connection Check** | **Verifies that the database is accessible, logging successes or failures and triggering alerts if the database connection fails.** | |
| * **API Health Check** | **Monitors specified critical APIs (e.g., SquareSpace Order API) for response status and speed, alerting administrators if response time or status is outside acceptable parameters.** | |
| * **Server Uptime Check** | **Calculates and logs the server's uptime, providing insights into the platform's stability and operational consistency.** | |
| * **Server Resouce Monitoring** | **Assesses CPU, memory, and disk usage, comparing current usage against defined thresholds, and triggers alerts if usage exceeds the acceptable limits.** | |
| * **Periodic Health Checks** | **Aggregates individual health checks (database, APIs, uptime, and resources) into a comprehensive status report and logs the overall system health.** | |
| * **Health Check Scheduler** | **Automatically initiates health checks at defined intervals (e.g., every 5 minutes), maintaining regular monitoring without manual intervention.** | |
| * **Adin API Endpoint (/admin/status)** | **Provides an HTTP endpoint for retrieving the current health status of the platform, responding with 200 OK if the system is healthy, or 503 Service Unavailable if issues are detected.** | |
| **Files it Gets Information From:** | | **Files it Sends too:** |
| * **Environment Variables** (.env): For database credentials and API URLs (SQUARESPACE\_ORDER\_API\_URL). | * **Logging File**: Logs to site\_status.log, recording all health check activities, resource status, and alerts triggered. | |
| * **Database (MySQL)**: Connects to the main Razda Market database for connection health verification. | * **Alert Manager**: Communicates with alert\_manager.py to trigger alerts on issues with database connectivity, API failures, or high resource usage. | |
| * **External APIs**: Makes requests to critical external services (e.g., SquareSpace) to ensure they are operational. | * **HTTP API Endpoint (/admin/status)**: Exposes a JSON-based endpoint for admins to review the real-time health status of the site. | |
| **Expected input into file:** | | **Expected output from file:** |
| **Environment Variables**: Includes sensitive information for database and API configuration.  ‘’’python  DB\_HOST, DB\_USER, DB\_PASSWORD, DB\_NAME, SQUARESPACE\_ORDER\_API\_URL, CPU\_THRESHOLD, MEMORY\_THRESHOLD, DISK\_THRESHOLD, RESPONSE\_TIME\_THRESHOLD  ‘’’  **User Input (CLI)**: Used indirectly for initiating health checks on schedule. | | **Log Records**: Stores log entries for each check performed, documenting successes, failures, and any triggered alerts.  ‘’’python  logging.info("Health Check Completed. Overall Status: OK")  ‘’’’  **API Output**: JSON response at /admin/status summarizing site health across multiple dimensions (database, APIs, resources). |
| **Things that need to be taking place:** | | |
| |  | | --- | | **● Database Connectivity: Ensures uninterrupted access to the main MySQL database, notifying admins of any connectivity issues that could affect platform functionality.** |  |  | | --- | | **● API Monitoring: Actively checks the availability and performance of essential APIs (such as SquareSpace Order API), which are critical for order processing and external integrations.** |  |  | | --- | | **● Resource Usage Thresholds: Maintains server performance by monitoring CPU, memory, and disk usage, automatically alerting admins if usage levels exceed predefined thresholds.** |  |  | | --- | | **● Scheduled Health Checks: Executes health checks at regular intervals using a background thread, guaranteeing continuous monitoring even if admins are not actively checking the status.** |  |  | | --- | | **● Alert System Integration: Works closely with alert\_manager to send alerts on critical failures (database, API, or server resources), enhancing the response time for incident management.** |  |  | | --- | | **● Uptime and Performance Logging: Tracks and logs server uptime and resource levels to identify long-term trends, enabling proactive maintenance planning.** |  |  | | --- | | **● Real-Time Health Reporting: Through the /admin/status endpoint, provides up-to-date system health insights, making it easier for admins to assess the system’s status and respond accordingly.** |  |  | | --- | |  | | | |
| Edit log (update each time you make changes to doc or file). | | |
| Contribution to the entire Project:  The **site\_status.py** file is a critical component for the operational reliability of the Razda Market platform. Its continuous monitoring of database connections, API health, and server resources ensures that admins receive prompt notifications of any potential issues, thus minimizing downtime and enhancing user experience. By automating health checks and incorporating real-time alerts, this module reduces the manual monitoring burden on administrators, enabling them to focus on more strategic tasks while maintaining site stability.  The /admin/status endpoint also provides an on-demand health check interface for administrators, giving them immediate visibility into the platform’s current status. This integration of automated monitoring, alerting, and real-time reporting positions the site\_status.py file as a cornerstone of the platform's infrastructure management, contributing to both the resilience and scalability of Razda Market.  Feel free to ask if you'd like to continue with another file or have any specific questions about this one. | | |
| * - Oliver Smith (Razda Admin) Nov 8, 2024: Established periodic health checks for database, APIs, and resources. Added logging and alert integration for real-time incident tracking. | | |